

ARE HUMAN ISSUES IMPORTANT IN LEGACY SYSTEM REENGINEERING?

Various quotes depict reengineering simply as an evolution from something old to something new:

"The systematic transformation of an existing system into a new form to realize quality improvements in operation, system capability, functionality, performance, or evolvability at a lower cost, schedule, or risk to the customer." [Reengineering Center, SEI, 95]

"Software reengineering is important for putting high software maintenance costs under control, recovering existing software assets, and establishing a base for future software evolution. The idea to improve, or transform, existing software so one can understand, control, and use it anew. Software reengineering is integral for achieving many goals in software maintenance and for planning for change in existing systems." [Arnold 93]

"System evolution is a coarser grained, higher level, structural form of change that makes the software systems qualitatively easier to maintain. Evolution allows the system to comply with broad new requirements and gain whole new capabilities. Instead of changing software only at the level of instructions in a higher level programming language, change is made at the architectural level. System evolution increases the strategic and economic value of the software by making it easier to integrate with other software and making it more of an asset than a liability. " [Weiderman 97]

“Understanding the old process and customer requirements, so as to recognize the weaknesses of the existing process and the performance demanded of the new one.” [Hamner 95]

These “definitions” almost seem to describe the evolution of something material. Legacy system software, especially Management Information Systems (MIS) interact with a great amount of human interface. They also were designed with a great amount of human interaction. Today's software lends itself to a more rapid self-generating development, but the software of 20 – 30 years ago was developed with tedious human intervention. My organization deals in MIS, which I imagine is unlike weapon system or space system software that interacts mainly with other software or hardware. Based on what I have experienced in my organization concerning reengineering projects, I feel that we've left out some of the human factors in our plans.

In the SEI publication “Why Reengineering Projects Fail”, the authors give the top ten reasons that reengineering projects fail as:

“1. The organization inadvertently adopts a flawed or incomplete reengineering strategy.

[We have a bulletproof strategy.]

2. The organization makes inappropriate use of outside consultants and outside contractors.

[We rely on experts to help us get there.]

3. The work force is tied to old technologies with inadequate training programs.

[We're known for our on-the-job training.]

4. The organization does not have its legacy system under control.

[We are on top of it—we know the system inside and out.]

5. There is too little elicitation and validation of requirements.

[Our needs are simple and straightforward.]

6. Software architecture is not a primary reengineering consideration.

[Anybody can specify an architecture.]

7. There is no notion of a separate and distinct “reengineering process.”

[We have our best people working on it.]

8. There is inadequate planning or inadequate resolve to follow the plans.

[We’re too busy to plan.]

9. Management lacks long-term commitment.

[Tomorrow is another day.]

10. Management predetermines technical decisions.

[If there’s one thing we’re good at, it’s giving orders.]” [Bergey Apr 99]

This paper will highlight 2,3, 9 and 10 as the failures dealing with human issues my organization has experienced.

Failure number 2 deals with the use of consultants and contractors. This trend has dramatically increased over the past 15 years due partially to the downsizing of the DOD workforce. Before this time, DOD had in-house software engineering training programs, and trainees gained experience developing and maintaining systems along with their more experienced counterparts. Most civilians remained at the same location for many years. DOD has chosen to downsize the workforce and has not brought any trainees in for quite some time. A majority of the personnel, whom were around when the legacy systems were developed, have or will be retiring. This has now left a void in legacy system understanding.

According to the Software Technology Support Center, Reengineering Technology Report, some of the reasons to reengineer include improving software maintainability, reducing system life cycle costs, extending software's life expectancy, building a foundation for quick adaptation to change and upgrading to newer technologies. A great number of DOD systems have reached a point of extremely high maintenance costs. The speed of the release of new computer software and hardware continues to increase dramatically, causing a greater imbalance in the cost of maintaining legacy mainframe systems.

DOD systems are in need of reengineering or replacement, but there is not a sufficient organic staff in place to execute this. Consultants and contractors are often brought in to either completely work on a project or as additions to the organic staff. They can often offer benefits to a project such as understanding of the domain, technical expertise, objectivity, or the ability to bring extra personnel to a project quickly. However, they can also contribute to the failure of projects. Outside contractors often have conflicting interests. The government wants to minimize the cost of external resources, while to their benefit; the contractor usually wants to maximize it. Many times they do not provide the experienced personnel they claimed they had or could recruit. There is also a high turnover with contractors and this makes it difficult to build up any experience on DOD systems.

The problems that downsizing have created have been exaggerated by the lack of training for organic personnel, which is failure number 3. Often times the existing organic personnel are so tied up supporting the legacy systems that they are not permitted the time to engage in training in new technologies. Other times the organization might feel as if new technologies can be picked up in a book or on the job. In the past, technology did not change as rapidly and

adapting to the new could be done at a slower pace. There are also those personnel that become comfortable and routine with their jobs and do not wish to expend energy learning anything new.

In all the references used for this paper, it is emphasized that in planning training, organizations need to consider the technical skills required for adopting new technologies, while also addressing motivational issues. Motivational issues include potential resistance to change. Management should present its rationale for the changes, as well as the potential benefits for employees in acquiring and using the new skills.

An important quality that an organization requires in order to successfully implement change is to have change sponsors at all levels of management. Sponsors allocate resources for change, remove barriers to change and provide ongoing reinforcement and encouragement. They will set the tone for change and provide incentive for those “bull headed” employees.

“The Reengineering Revolution” [Hamner, 95], which is a book discussing organizational reengineering, emphasizes the need for change sponsors from the topmost levels of the organization. No matter how committed or how hard they try, the personnel on the front lines are in no position to launch and sustain a major project. The front line personnel lack the breadth of perspective to see a project from start to finish rather than their own narrow expertise. Front liners might think that collectively they can be innovative, but they will soon realize that they can do nothing without management who control resources and translate policy into practice. True empowerment is usually a politically correct term.

In conjunction with being change sponsors, management must support the reengineering project until it is completed or failure number 9 comes in. Management support of the project will require careful monitoring and putting things back on track when they stray. If management becomes distracted by other projects during the course of a major reengineering effort, they will not know when things go wrong. An important issue as it relates to software engineering is that the consequences of not correcting problems in the early to middle phases of the lifecycle can be catastrophic because errors are harder and more costly to correct if they are found late in the process. It often seems that management gets pulled off to other "hot" projects or moved to another organization and then they leave the previous project to lose its momentum.

Failure number 10 is an issue that I have experienced on more than one occasion in my organization. This issue can be seen in new development, reengineering and maintenance projects. Problems occur when upper management mandates or edicts the technical approach, schedule, cost, or performance considerations without sufficient project team input or concurrence. Dealing with software is not an easy business, made more difficult by working with legacy systems that have poorly developed components and poor documentation, projects with fluid requirements and dysfunctional organizations. The references used for this paper explain that while top management's job is to make decisions on the allocation of scarce resources, it is also tempting for them to determine specific deliverables and timetables, and to make promises to customers that cannot be kept. In a reengineering project, the detailed planning of schedules and milestones can only be accurately determined through careful study of the technical nuances of a system, based on an understanding of the system, historical data, and knowledge of the specific skills of the staff. When top management puts their hands in the

details without basing their decisions on the technical analysis of others, the results are usually not good.

Human issues are important to legacy system reengineering. In order for a reengineering project to be successful, these need to be addressed in goal setting, planning, the determination of organizational readiness and management commitment. If they are not addressed, then project failure is a most likely option.

References

[Arnold 93] Arnold, Robert S, Software Reengineering, IEEE Computer Society Press, Los Alamitos, CA 1993.

[Bergey Apr 99] Bergey, John; Smith, Dennis; Tilley, Scott; Weiderman, Nelson; Woods, Steven; Why Reengineering Projects Fail; April 1999

[Bergey Sep 99] Bergey, John; Smith, Dennis; Weiderman, Nelson; DoD Legacy System Migration Guidelines; September 1999

[Hamner 95] Hamner, Michael; Stanton, Steven; The Reengineering Revolution; 1995.

[Olsem 93] Olsem, Michael R.; Sittenauer, Chris; STSC Reengineering Technology Report, Volume 1; August 1993.

[Reengineering Center, Software Engineering Institute, 95] Perspectives on Legacy System Reengineering, 1995.

[Weiderman 97] Weiderman, Nelson; Northrop, Linda; Smith, Dennis; Tilley, Scott; & Wallnau, Kurt. Implications of Distributed Object Computing for Reengineering, 1997.